

Device-Free, Activity during Daily Life, Recognition Using a Low-Cost Lidar

Zixiang Ma, John Bigham, Stefan Poslad, Bang Wu, Xiaoshuai Zhang, and Eliane Bodanese

School of Electronic Engineering and Computer Science, Queen Mary University of London, London, United Kingdom

Email: {zixiang.ma, John.bigham, stefan.poslad, bang.wu, xiaoshuai.zhang, eliane.bodanese}@qmul.ac.uk

Abstract—Device-free or off-body sensing methods, such as Lidar, can be used for location-driven Activities during Daily Life (ADL) recognition without the need for a mobile host such as a human or robot to use on-body location sensors. Because if such an attachment fails, or is not operational (powered up), when such mobile hosts are device free, it still works. Hence, this paper proposes an innovative method for recognizing ADLs using a state-of-art seq2seq Recurrent Neural Network (RNN) model to classify centimeter level accurate location data from a low-cost, 360° rotating 2D Lidar device. We researched, developed, deployed and validated the system. The results indicate that it can provide a centimeter-level localization accuracy of 88% when recognizing 17 targeted location-related daily activities.

Keywords—Lidar; Location-based Services; Seq2seq; Activities during Daily Life (ADL); Robot localization; Recurrent Neural Network (RNN); Smart Home; Healthcare;

I. BACKGROUND AND RELATED WORK

This paper studies the recognition of a mobile host's activities during daily life (ADL) solely based upon location tracking using a low-cost 2D Lidar. The mobile host or subject could be a human or a robot; activities could be at home or work. No sensors are worn by or embedded into, the subject – it is device free. Other sensors, e.g., inertial sensors, could be deployed to assist the recognition of ADLs, but *our focus is on location driven ADLs*. Our initial target location space is a small building where high location accuracy is necessary to discriminate between which landmarks therein the host is nearby to. This is of interest as it tends to be used for example for: sheltered human accommodation, i.e., for the elderly, those in rehabilitation and those recovering from severe depression; in a future where robots are used ever more for health care, to handle hazardous materials and to monitor remote pipelines [1]. Today there are several Indoor Position Systems (IPS) that support applications such as navigation, proximity monitoring, and emergency location systems. These have been developed because of the lack of the Global Navigation Satellite (Positioning) System (GNSS) signal indoors [2]. The most popular IPS is WiFi, as it can make use of an already deployed Wireless Access Point (AP) infrastructure to determine the position. Similarly, Bluetooth is another widely used inexpensive wireless IPS technology that is energy efficient. Bluetooth enabled receivers, such as smartphones can pick up these signals and respond accordingly when a Bluetooth beacon comes into range. This operation can also be inverted, where the subject wears a beacon, and other fixed devices in the home environment receive the beacon signal strength and other

sensor data from the person. Both WiFi and Bluetooth estimate location using forms of trilateration or by constructing a radio map prior to the location estimation and then matching the current signals to a set of locations in the radio map index using the Received Signal Strength Indicator (RSSI). Inside a small home, trilateration does not provide good accuracy because of non-line-of-site issues. Radio maps mitigate these inaccuracies, but due to signal interference for both WiFi and BLE, the estimates still limit the accuracy, typically to 1-2 m which is too inaccurate. This typically needs to be 1 to 2 orders of magnitude lower to differentiate physical items, such as sink, fridge or kettle use in a small building. The use of WiFi and Bluetooth require the host to carry an on-body device. Some studies have shown that some target users such as seniors are more disinclined to use such eHealth wearables [3]. Note, such attached devices, may require a degree of maintenance such as recharging batteries to keep them operational.

Much research has investigated how to improve indoor positioning accuracy, e.g., [4], [5], [6], [7] demonstrates the benefits of using a hybrid approach. [4] deploys Bluetooth in the home using inverted RSSI fingerprinting, step counting, magnetometers and, importantly, focuses on detection of moving path segments (using dynamic time warping) combined with stationary waits or stays referred to as Stay Points. This has achieved a good accuracy for recognising elementary actions. This work also suggests an activity-centric trajectory based prediction model as a practical solution for location estimation in homes. This is achieved by populating the radio map using sequential RSSI data collected along a path (e.g., couch to the dining area) and stationary landmarks (e.g., sleeping on a bed), that are associated with an ADL. However, such a multi-sensor IPS still suffers from the same limitations as the WiFi and Bluetooth methods in terms of insufficient location accuracy and requiring the use of wearables.

Another IPS choice is UWB, which has a high accuracy of a few centimeters. UWB devices use a very large bandwidth and can transmit high data rates over short distances at very low power levels. They are not affected by the existence of other communication devices, or external noise [8], although, they can be affected by other wide spectrum devices if misconfigured, e.g., WiMAX and digital TV [9]. UWB location estimation is based on Time difference of arrival (TDOA)-based algorithms or Time of arrival or flight (TOA)-based algorithms. The use of narrow pulses makes UWB very tolerant to multipath effects. However, despite the accuracy, it does require the mobile host to carry a location transmitter.

The final type of off-body sensor considered is light-based: various cameras or Light detection and ranging (Lidar), which uses pulsed laser light and measures the TOA reflected pulses with a sensor. Major weaknesses of cameras are that they do not accurately determine the location and are highly (human) privacy-invasive. A range of Lidar devices could be used for ADL detection such as flash Lidars that face in a single direction (1D), line scanning sensors that swept a beam across a scene, measuring along a single (2D) plane and 3D Lidar [10]. N.B., both 2D and 3D Lidar devices can be designed to rotate 360 degrees. The major application of Lidar is for mobile unmanned vehicles and robots to track them as they move [10], [11] and to support collision avoidance. *To the best of our knowledge, no work has so far looked at using device-free low-cost (2D rather than 3D) Lidar devices to recognise ADLs.*

In addition, to selecting and deploying sensors to accurately measure hosts' locations, we also need to consider how to analyse any associated activities derived from this. Understanding the activities of a mobile host requires contextual information related to their inherent surroundings to be recorded. One approach to model ADLs is based on a task-specific and intention-oriented plan representation language such as Asbru [12] and used for modeling and applying medical protocols. [13] developed a recognition engine to detect ADLs that were modeled using Asbru from sensor events, principally RFID tags. Another way of representing and modeling high tier behavior is workflows, i.e., to use augmented Petri Nets [14]. However, workflows are too prescriptive in their ordering. If workflows are applied in dynamically changing environments, they require a large number of permutations to be explicitly enumerated. Workflows can scale poorly to cases where there are many possibilities [15]. In addition to scalability issues, it can be tough to manage the representation of priorities and ordering. Thus, more flexibility is required when modeling hierarchical ADLs, including the variable duration of the same task, variable ordering of tasks, and the degree of overlap with other ADLs. Techniques which attempt to map these as a flat structure are problematic because they are unable to model flexible scenarios, such as interweaving ADLs. The ability to support interweaving ADLs is also a crucial advantage [16].

ADL recognition often requires the enumeration of hierarchical plans for each ADL, i.e., a library of ADLs has to be created, where each ADL is elaborated. Another approach is to bypass such a process using an inductive approach using patterns in behaviors that are labeled. [17] has used a pattern mining approach where the emphasis was in detecting unexpected behaviors. *In this paper, we, however, attempt to detect normal patterns using a Recurrent Neural Network and to depend on Lidar data as the only sensor data.*

The main contributions and novelty of this paper are 4-fold:

- 1) *We propose an accurate 2D Lidar-based device-free location driven ADL recognition method.*
- We further extend this method so that we:
- 2) *Classify multiple mobile hosts in real time, as there may be more than one user in the same space;*

- 3) *Propose a state-of-art seq2seq model to analyse the Lidar data, without needing to enumerate hierarchical plans for each ADL to improve the detection accuracy;*

- 4) *As location-driven ADLs may not have a mobile host presence at specific locations for much of the time, leading to much recorded data often containing no ADLs and a massive largely redundant dataset, we propose to detect the presence of moving mobile hosts to avoid this.*

The remainder of this paper is organized as follows: Section II introduces the methods we have adopted and describes the experiments we performed. Section III presents the performance of our seq2seq-based activity recognition method. Finally, Section IV concludes with a summary of the results of our experiments and proposes future work.

II. METHOD

Our system consists of two phases: first, is 2D Lidar-based indoor localization, which will be used for recognition activity; Second, is activity recognition that uses the processed location data as the inputs to our seq2seq model.

A. 2D Lidar Localization and Multi-user Tracking

1) Lidar Data Collection

A low cost, 2D, rotating Lidar system called RPLIDAR (version A1) from Slamtech¹ was used and configured to continuously scan a physical space until the user switches it off. 2D Lidar is height sensitive to what it can detect. A good height for horizontal pulses is to hit the upper mobile host, e.g., human torso, and to have a direct line of sight to all the indoor landmarks of interest, else the ADLs at these landmarks will be missed. Lidar uses a type of laser to emit the light pulses to detect distance. This is a Category 1 laser², the safest category, which cannot emit levels of optical radiation above the exposure limits for the human eye under any exposure conditions.

For many location-driven ADLS, there is often no mobile host presence in a space for much of the day. Hence, much recorded data often contains no ADLs leading to a massive, largely redundant, data set. Hence, we use a threshold based on Hausdorff distance, which measures how far two subsets of a metric space are from each other, representing two time sequential scans, to detect the presence of a moving user.

First, we convert the raw Lidar radial distance and angle data to Cartesian coordinates. The position of Lidar is denoted by the origin point (0, 0). Then the Hausdorff distance between different scans I and J is calculated. The Hausdorff distance between Lidar scans I and J is defined as:

$$H_d(I, J) = \max\{s_{i \in I} i_{j \in J} d(i, j), s_{j \in J} j_{i \in I} d(i, j)\} \quad (1)$$

¹ Available from <https://www.slamtec.com/en/Lidar/A1>.

² IEC-60825-1: Safety of laser products International Electrotechnical Commission. Edition 1.2, August 2001

where s represents the supremum, i represents infimum. $d(x, y)$ means the Euclidean distance between a point i in scan I and point j in scan J .

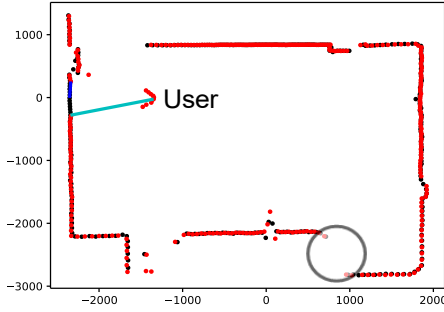


Figure 1. How the Hausdorff distance works.

The origin point (0,0) is where the Lidar is located, and the distance measuring unit is millimeters. The black points are from a scan (e.g., scan 1), without any moving objects. The red points are from a scan (scan 27) where a user walks into the kitchen. Mobile host movement is what causes the blue lines. The length of the light blue line shown in Figure 1 means $\sup_{x \in X} \inf_{y \in Y} d(\text{scan}27, \text{scan}1)$, which is also the directed (or forward) Hausdorff distance between scan 27 and scan 1. The distance of the dark blue line equals directed Hausdorff distance between scan 1 and scan 27 - $\sup_{x \in X} \inf_{y \in Y} d(\text{scan}1, \text{scan}27)$. When a mobile host moves, the kitchen boundary beyond the user from the Lidar device is not detected as the mobile host blocks the light beam. Instead of using the Hausdorff distance, we also tried to differentiate between two scans based on comparing distances at each angle. However, a small angle difference can lead to a considerable difference in distance, which is shown in Figure 1 (the grey circle).

2) 2D Lidar Data Pre-processing

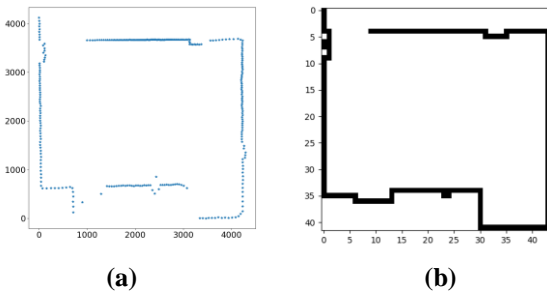


Figure 2. Building a grid map from Lidar point data where each grid equals $10 \times 10 \text{ cm}^2$.

3) Density-based spatial clustering of applications with noise

On occasion when using Lidar, a few outliers will also show in our processed location-based ‘image map.’ This would reduce the localization and recognition accuracy, especially when more than one mobile user is present in the same room. So, we used Density-based Spatial Clustering of Applications

with Noise (DBSCAN) to reduce such anomalies and to represent different mobile hosts as clusters.

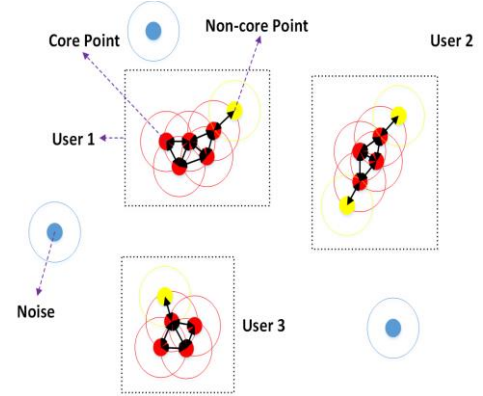


Figure 3. Three clusters are generated (for minPts=2).

DBSCAN is a clustering algorithm based on regional data density, which is often used for outlier detection. It also marks outliers lying in low-density regions for removal. The algorithm requires two parameters: radius ϵ , minPts. Points are classified as follows:

- A core point means at least minPts points are within a ϵ radius. Each cluster consists of at least one core point.
- A non-core point means the point we can directly reach is less than minPts core points. We can count non-core points as part of the same cluster, or delete them as they cannot be used to reach more points.
- All points that are not reachable from any other point are considered as noise points (outliers).

B. Human Daily Activities Recognition

1) Recurrent Neural Networks (RNN)

Our target is to recognise location-related human daily activities, the inputs are sequential locations, in our case, 2D Lidar data. A natural choice to classify sequential data is to use a Recurrent Neural Network (RNN). We selected Long short-term memory (LSTM)³ as our baseline algorithm to recognise activities, which is one of the most used RNNs.

Then the research question for our project is how to build the corresponding output activities, e.g., a kitchen has various landmarks, such as a table, sink, fridge, and microwave. Lidar can be used to track motion between landmarks and the continued human presence at a landmark – a so-called stay point. One solution to define those output activities can be based on those landmarks, as most activities happen near those landmarks. However, there could be different landmarks in different users’ houses. So we use a Stay Point Recognition algorithm to recognise those key landmarks.

2) Stay Point Recognition

³ We used Keras to implement the LSTM that is easy to use. It’s available from <https://keras.io/>.

We first segment the data into fixed time periods of 30 seconds and then identify points where the host stays for at least a few seconds, so-called stay points. 30s was chosen as a trade-off between the time taken to move between landmarks, and to allow sufficient time to link and analyse a small chain of landmarks visits that may be stay points together as simple ADLs, versus creating and having to analyse much more complex ADL chains. We use a stay point recognition algorithm [18] to identify key landmarks in this time segment, as our basic assumption is that staying at a location means that this point is significant. The subject may well pass other landmarks, but these are not necessarily relevant to the activity being performed.

The basic idea for stay point recognition is as follows:

1. The trajectory is represented as the set:

$$S = \{loc_k = (t_k, l_k) | k = 1, 2, \dots, n\} \quad (1)$$

where l is the Cartesian coordinate set, t is the timestamp.

2. We transform the data into the following equations:

$$f(t_k) = t_k - t_e \quad (2)$$

$$Z = g(f(t_k)) = \sum_{n=2}^k distance(l_n, l_{n-1}) \quad (3)$$

t_e is the time when a mobile host, e.g., enters a kitchen, and distance is the Euclidean distance. The horizontal axis in Figure 4 represents $f(t_k)$ is the time when the host enters the kitchen and $distance(l_k, l_{k-1})$ is the Euclidean distance between two points. The horizontal axis in Figure 4 represents $f(t_k)$ and the vertical axis represents Z , which is a monatomic increasing curve. In equation (3), we subtract the time of entry because no data collection is logged when the kitchen is empty.

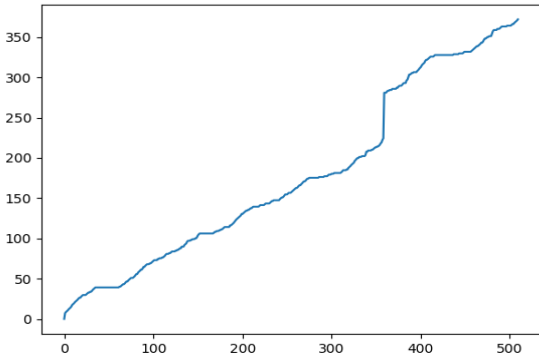


Figure 4. After transformation to function Z .

As $g(l_k)$ increases, its first derivative will be always greater or equal to zero. The stay points should be where g is horizontal, i.e., it is the local minima of the derivative of the curve. To identify the local minima, we check the zero-crossing of the second derivative, as the second derivative may not equal to zero as g is a discrete function. Figure 5 depicts the second derivative, and the red boxes means the selected regions of interest. They not only include stay points but also include

inflections, which are caused when a mobile host moves slowly, then speeds up.

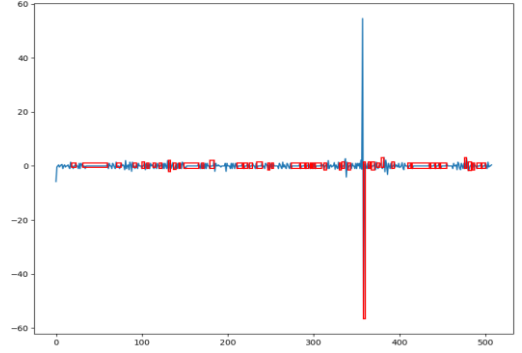


Figure 5. The second derivative of the curve.

To reduce inflections, we also use a confidence value to recognise the stay point. Based on paper [18], we also define a single confidence value of a data point as:

$$C(P_i) = 100 \text{ } g' \leq 0.01 \quad (4)$$

$$\text{Else: } C(P_i) = 100 - \frac{g' - 0.01}{0.01} \quad (5)$$

$$C(P_k, P_w) = \frac{\sum_{i=k}^w C(P_i)}{w - k + 1} \quad (6)$$

where g' is the first derivative. We set $C(P_k, P_w)$ in $C(P_i, P_j)$. i, j are the boundary points of zero-crossing of region k . w is where the confidence level of this sub-region is above 80. The center of those points will be treated as a stay point.

3) Sequence-to-sequence Model

Seq2seq is a general-purpose encoder-decoder model, which is initially built for machine translation [19], but that has also been used for a wide variety of tasks, e.g., text summarization. The reason that we chose to use a seq2seq model is that we thought the sequential location information could be analogous to a translation problem. Each sequential location information belongs to different corresponding activities.

Because activities can take different amounts of time, selecting fixed time periods to detect activities, i.e. 30s, then mapping each 30s period to just one activity may not work. To collect the training data, we segmented the 30 seconds using stay points and then annotated the stay points and annotated sub-sections of each segment with elementary activities. To validate our method, Lidar data was collected in part of a building, e.g., a kitchen. We asked two human mobile hosts to label the stay points and their related activities. The trajectory of every 30s is used as the input data to train the seq2seq2 model. The corresponding activity will be the labeled data, e.g., in an example 30s time slot, the corresponding activities are {from table to fridge, use fridge, prepare to cook}, then those activities can be represented as {2, 3, 8} (see the list given in Section III), then based on the activity, identifier we converted these into a one-hot encoding, where categorical variables are

converted into a 0 or 1 form, which then forms the input into the seq2seq model. After this transformation, more sequential training data is generated to train our model.

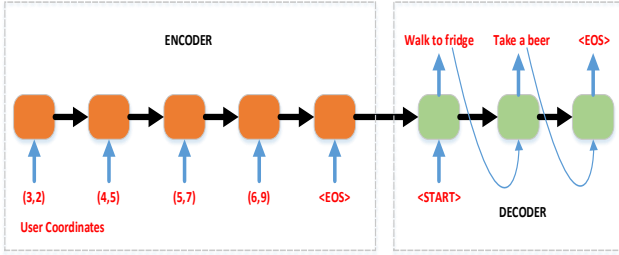


Figure 6. The sequential coordinates will be the input of the encoder; then the decoder will output the corresponding activities, where <EOS> means the end of the sequence.

III. EVALUATION

A. Experimental Setup

The dataset we collected and analysed consisted of a total of 536 times 30s time slots of acquired Lidar data, collected for 2 human mobile hosts. Of this, we used 100 times 30s for the test set and the rest for the training set. The experiment area is a kitchen as shown in Figure 2 and covers about 20 m². The Lidar we used is a 2D RPLIDAR, which is shown in Figure 7. It supports a 5-10Hz Adaptive Scan Frequency, 0.2 to 10 m (in our test) distance range, and is much lower cost compared to older rotating 2D Lidar devices.



Figure 7. The 2D RPLIDAR device.

B. Human Activity Recognition Performance

For the localization part, the system works well. Based on the comparison, each data point can be classified in a 2D grid. However, if two mobile hosts are close enough (less than 20 cm apart), two clusters may be merged into one, but this was not a big issue. A mobile host that is detected by the system could become undetected within a later period in the same sequence, that did not leave via a door stay point. We could then perhaps assume this has changed its vertical position, e.g., fallen, as the Lidar scan is set to a certain height (1.1m in our case). 1.1 m is chosen as a trade-off between being low enough to detect kitchen units that are offset from the wall so we can detect we are at them versus being high enough to detect an upright human. If a user becomes undetected, at the height of less than 1.1m for a period during a sequence, we could presume the mobile host has crashed towards the floor.

For human activity recognition, we use the stay point as the reduced set of important landmarks and as an activity

connector. So, the first step is that we recognize a mobile host's stay points, then ask an observer to label the stay points and the activity category for the ground truth. The next step is to use the sequential location data as inputs to train the model, where the corresponding label data represents the labeled activities (see figure 6).

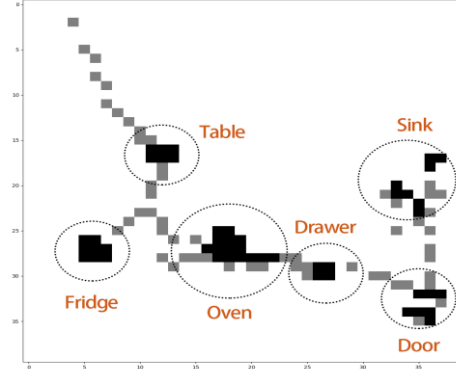


Figure 8. Labelled stay points

In Figure 8, the grey grids represent a user's trajectory over a period. The black ones mean the recognised stay points. As an example, we only visualise part of the dataset. Here, six stay points (landmarks) are identified. We asked an observer to label those stay points and to point out the related activities to provide the ground truth. There are several ways to label ADLS. First, we set several elementary ADLs, i.e., eating, drinking or use the fridge, then ask the user to label the corresponding sequential location data. Another way is to ask the user to label personalized activities. In our experiment, there are 17 activities labeled by a mobile host based upon daily simple kitchen activities. These are: 1) from door to table, 2) from table to fridge, 3) use fridge, 4) fridge to food pantry, 5) prepare cat food, 6) food pantry to back door, 7) feed the cat, 8) no one in the kitchen, 9) open back door, 10) back door to fridge, 11) prepare to cook, 12) oven to fridge, 13) fridge to back door, 14) use breadboard (drawer), 15) use oven, 16) go out of back door, 17) washup.

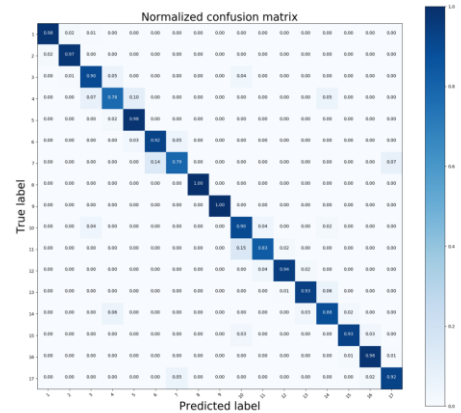


Figure 9. Recognition confusion matrix

Note that the labeled activities can include several different hierarchical levels. For example, we recognize 'food pantry' as

a stay point. Within a specific 30s, a human user can either move from the food pantry to backdoor or stay at or very near to the food pantry. Actions can also be labelled differently. A mobile host is not only near a stay point but does ADLs associated with a stay point, e.g., preparing cat food in the food pantry or putting something into the food pantry.

Although the sequence recognition accuracy is reasonable at 88%, when the predicted sequences are the same as the ground truth ones, a few activities from the total activity sequence are wrongly recognised, which is shown in Figure 9. The darker the blue, the more accurate the recognition. 1 to 17 represent the 17 labelled activities. Our hypothesis is that those activities that involve interactions with fewer, clearer to detect stay-points, can achieve a better recognition accuracy than those activities with more stay points that can be linked to multiple ADLs.

IV. CONCLUSION

Traditional ADL recognition usually requires mobile hosts to carry sensors to generate the data that is analysed via an enumeration of hierarchical plans. In contrast, we proposed, developed and validated 2D Lidar measurements of ADLs (used by two different people at different times). We combined the use of Lidar with the state-of-art seq2seq RNN model to classify ADLs linked to the Lidar generated stay points and transitions between stay points. Our validation shows that a 2D Lidar location determination method can provide centimeter-level localization accuracy and a good accuracy (88%) in recognizing seventeen location-related daily activities.

Our future research extensions are as follows. First, instead of using sequential location inputs, we will investigate the use of sequential landmarks as stay points in the training data, which may make the system more adaptive to different scenarios. Second, we will extend the evaluation of our system in different ADL scenarios that involve more mobile users. Third, we aim to investigate the application of other state-of-art zero-shot/one-shot learning to use less samples to train the system. Fourth, we will combine our previous research using RSSI localization methods [6], [20] to raise the recognition accuracy. Fifth, we will look at more distributed AI algorithms to analyse ADL data [21], [22]. Finally, we will investigate Lidar use more for robotic activity, including its use to recognise combined human and robot activities in a shared confined space as there is little work on the use of 2D Lidar for activity recognition of robots.

ACKNOWLEDGEMENTS

This research was funded in part by a China Scholarship Council and QMUL PhD Grant and by the UK EPSRC National Centre for Nuclear Robotics, NCNR project (EP/R02572X/1). We also gratefully acknowledge the support of NVIDIA Corporation who donated the GPU used for the data analytics.

REFERENCES

- [1] B. Gates, "A robot in every home," *Scientific American*, vol. 296, no. 1, pp. 58-65, 2007.
- [2] Z. Liang, I. Barakos and S. Poslad, "Indoor Location and Orientation Determination for Wireless Personal Area Networks," *Lecture Notes in Computer Science*, vol. 5801, pp 91-105, 2009.
- [3] M. Levine, S. R. Lipsitz, and J. A. Linder, "Trends in seniors use of digital health technology in the united states, 2011-2014," *Journal of the American Medical Association (JAMA)*, vol. 316, no. 5, pp. 538-540, 2016.
- [4] M. Sridharan, J. Bigham, C. Phillips, and E. Bodanese, "Collaborative location estimation for confined spaces using magnetic field and inverse beacon positioning," *IEEE Sensors*, 2017, pp. 1-3.
- [5] Z. Ma, S. Poslad, S. Hu, and X. Zhang, "A fast path matching algorithm for indoor positioning systems using magnetic field measurements," *28th IEEE Ann. Int. Symp. Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1-5.
- [6] Z. Ma, S. Poslad, J. Bigham, X. Zhang, and L. Men, "A BLE RSSI ranking based indoor positioning system for generic smartphones," *Wireless Telecommunications Symposium (WTS)*, 2017, pp. 1-8.
- [7] K. A. Nguyen, C. Watkins, and Z. Luo, "Co-location epidemic tracking on london public transports using low power mobile magnetometer," *arXiv preprint arXiv:1704.00148*, 2017.
- [8] S. Krishnan, P. Sharma, and O. H. Woon, "A uwb based localization system for indoor robot navigation," in *IEEE Int. Conf. on Ultra-Wideband, (ICUWB)*, 2007, pp. 77-82.
- [9] R. Aiello and A. Batra, "Ultra wideband systems: technologies and applications," *Newnes*, 2006.
- [10] J. Shackleton, B. and VanVoorst, "Tracking people with a 360-degree lidar," *7th IEEE Int. Conf. on. Advanced Video and Signal Based Surveillance (AVSS)*, 2010, pp. 420-426.
- [11] A. Wąsik et., "Lidar-based relative position estimation and tracking for multi-robot systems," *2nd Iberian Robotics Conf.*, 2016, pp. 3-16.
- [12] C. Fuchsberger, J. Hunter, and P. McCue, "Testing asbru guidelines and protocols for neonatal intensive care," *Conf. on Artificial Intelligence in Medicine in Europe*, 2005, pp. 101-110.
- [13] U. Naeem, J. Bigham, and J. Wang, "Recognising activities of daily life using hierarchical plans," *Eur. Conf. on Smart Sensing and Context*, Springer, 2007, pp. 175-189.
- [14] R. Zurawski, "Petri nets and industrial applications: A tutorial," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 6, pp. 567-583, 1994.
- [15] W. van der Aalst, "How to handle dynamic change and capture management information," *Int. Conf. on Cooperative Information Systems*, pp. 115-126. 1999.
- [16] S. Nasreen, M. A. Azam, U. Naeem, M. A. Ghazanfar, and A. Khalid, "Recognition framework for inferse2seqing activities of daily living based on pattern mining," *Arabian Journal for Science and Engineering*, vol. 41, no. 8, pp. 3113-3126, 2016.
- [17] U. Naeem and J. Bigham, "Activity recognition in the home using a hierarchal framework with object usage data," *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 4, pp. 335-350, 2009.
- [18] G. Stylianou, "Stay-point identification as curve extrema," *arXiv preprint arXiv:1701.06276*, 2017.
- [19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, 2014, pp. 3104-3112.
- [20] B. Wu, Z. Ma, S. Poslad. and W. Zhang, "An efficient wireless access point selection algorithm for location determination based on RSSI interval overlap degree determination," *Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1-8.
- [21] S. Poslad, P. Buckle and R.G. Hadingham, "FIPA-OS: the FIPA agent Platform available as Open Source." *5th Int. Conf on the Practical Application of intelligent Agents and Multi-agent technology (PAAM)* 2000, pp. 355-368.
- [22] S. Poslad, P. Buckle and R.G. Hadingham, "Open Source, Standards and Scaleable Agencies," *Lecture Notes in Computer Science*, vol. 1887, pp. 296-303, 2001.